UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/772,650	01/30/2001	Harm Sluiman	CA920000042US1	1018

61136 7590 03/30/2007

HAMILTON & TERRILE, LLP
P.O. BOX 203518
AUSTIN, TX 78720

EXAMINER	
KANG, INSUN	
ART UNIT	PAPER NUMBER
2193	
MAIL DATE	DELIVERY MODE
03/30/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/772,650
Filing Date: January 30, 2001
Appellant(s): SLUIMAN, HARM

Stephen A. Terrile

For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed on 10/6/2006 appealing from the Office action mailed on 3/23/2006.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

The following is a listing of the evidence (e.g., patents, publications, Official Notice, and admitted prior art) relied upon in the rejection of claims under appeal:

1) 6,633,888 Kobayashi 10-2003

2) APA (Applicant's admitted prior art) disclosed in the instant application.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,633,888 to Kobayashi in view of Applicant's Admitted Prior Art (hereinafter referred to as "APA") disclosed in the instant application.

The following ground(s) of rejection are applicable to the appealed claims (claims 1-8) and were set forth in the final Office action mailed on 3/23/2006.

Claim Rejections - 35 USC § 103

3. Claims 1-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent 6,633,888 to Kobayashi in view of Applicant's Admitted Prior Art (hereinafter referred to as "APA") disclosed in the instant application.

Per claim 1:

Kobayashi teaches:
- testing a software test component (i.e."testing newly created component classes within the visual builder interface," in col 4 lines 62)

- ascertaining a public interface of the software test component (i.e. "once the interface of a bean is known, a programmer can create a new customized component from the base Java bean component," col 7 lines 31-45; see also col 8 lines 33-58; col 8 lines 33-58; col 8 lines 33-58)

- creating a wrapper component for the software test component (i.e. "a proxy component is created for each method, including constructors," abstract) by the substeps of defining a wrapper component interface to mirror the public interface of the software test component (i.e. "the parser/extractor 304 parses each constructor and each method and extracts any related fields, comments, and parameter names," col. 8 lines 46-58 ; "proxy component encapsulates the parameters of that method. In particular, parameters associated with a method are represented by properties of the proxy component created for that method," col 5 lines 1-9)

- defining the wrapper component to delegate to the software test component and to receive calls to the software test component(i.e. "a proxy component is created for each method, including constructors," abstract ; "the bean compiler converts each component into proxy components," col 8 lines 8-19) by including calls to the public interface of the software test component within the wrapper component (i.e. "constructor and methods objects instantiated by the proxy beans 210 within bean-based application 216 to call the appropriate constructors and methods for the target class in the implementation code," col. 9 lines 40-54 ; "the methods of proxy beans are invoked, they use the universal transport mechanism to invoke the actual component code in order to test the method," col 22 lines 41-53; see also col 12 lines 18-25).

Kobayashi teaches that the "proxy components can be manipulated ...[and] Each composite component in the application can be tested...under control of the proxy components (col 8 lines 8-32; col. 22 lines 54-67 and col. 23 lines 1-6)." Although the proxy component can be edited to insert test code to capture and playback of user interaction with the interface, Kobayashi does not explicitly states capturing and playback of user interaction. APA discloses that such "GUI capture and playback tooling (page 1, specification)" was known in the art of software development and testing, at the time applicant's invention was made, to make "the recorded user-GUI

interfaction available for repeated test cases (page 1, specification)." It would have been obvious for one having ordinary skill in the pertinent art to modify Kobayashi's disclosed system to capture and playback user interactions disclosed in APA. The modification would be obvious because one having ordinary skill in the art would be motivated to record user-GUI interaction so that it can be used for repeated test cases (page 1, specification) as taught by APA.
- enabling a test case to use the wrapper component interface to pass the received calls to the software test component and to generate test data from the test code in the wrapper component (i.e. "when the methods of proxy beans are invoked, they use the universal transport mechanism to invoke the actual component code in order to test the method...the method parameters of the original bean are exposed by the proxy components created from the methods of that bean," col 22 lies 46-53).
substantially as claimed.

Per claim 2:
The rejection of claim 1 is incorporated, and further, Kobayashi teaches:
- the software test component is an object-oriented software test component (i.e. "The beans to be tested," col 22 lines 18-40)
- interrogating a test component definition to determine public methods, constructor and associated parameters for the software test component (i.e. "the parser/extractor ... parses each constructor and each method and extracts any related fields, comments, and parameter name," col 8 lines 33-58)
as claimed.

Per claim 3:
The rejection of claim 2 is incorporated, and further, Kobayashi teaches:
-the test component is a Java language class (i.e. "The beans to be tested," col 22 lines 18-40)

- use of an introspection group of interfaces in a Java Bean specification (i.e. "the parser/extractor ... parses each constructor and each method and extracts any related fields, comments, and parameter name," col 8 lines 33-58)
as claimed.

Per claim 4:
The rejection of claim 2 is incorporated, and further, Kobayashi teaches:
- defining public methods, constructors and associated parameters in the wrapper component to mirror the public methods, constructors and parameters determined for the software test component (i.e. "Using the extracted constructor information, the compiler module creates and compiles a constructor bean such as beans and ...The compiler ...also creates a method bean from extracted information for each method in the class," col 8 lines 33-58; "a proxy component is created for each method, including constructors ...which proxy component encapsulates the parameters of that method. In particular, parameters associated with a method are represented by properties of the proxy component created for that method," col 5 lines 1-9)
as claimed.

***Per claim 5**, this is the computer program product version of claim 1, respectively, and is rejected for the same reasons set forth in connection with the rejection of claim 1 above.*

***Per claims 6-8**, they are the system versions of claims 1, 2 and 4, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1, 2 and 4 above.*

(10) Response to Argument

As an initial matter, the above ground(s) of rejection were set forth in the final office action mailed on 3/23/2006 and are reproduced above at pages 3-6 of this examiner's answer.

1) Appellants contend that: Merely because the limitations of the claims are possible through features in the Java Bean specification does not mean that this same specification discloses or suggests such limitations. As discussed in the previous response, Applicant has found a clever way to tap into the data action during testing. The mirror interface allows the wrapper to support calls to access the interface of the software test component. Data is not changed to overcome an incompatibility, it is collected for evaluation. These features are not disclosed or suggested by Kobayashi or the APA (brief, page 4, last paragraph).

In response, the present invention uses reflection (mirror interface) in the Java Bean specification to generate the wrapper (proxy, delegate) by gathering the interface of the actual component and introspecting the details of the actual component (i.e. instant specification, page 7 last paragraph). Reflection is the return of an image in a mirror to discover information about Java classes (introspection). The methods from reflect package and Class class hold a mirror up to a class to look inside to find information about the classes fields, methods and events. Kobayasi clearly discloses defining proxy components to mirror and therefore delegate the actual components (i.e. "the bean compiler 208 converts each component into proxy components 210...the underlying objects and components 202 to be exercised under control of the proxy components," col. 8 lines 11-18 and 23-27) by using the parsing/extracting mechanism

(i.e. col. 9 lines 5-19). The parsing/extracting mechanism is to determine/describe (i.e. introspection) and obtain (i.e. reflection) information about the members of a class such as the properties, methods, and constructors (i.e. "the parser/extractor 304 parses each constructor and each method and extracts any related fields, comments, and parameter names," col. 8 lines 46-58). This extracting mechanism extends the "conventional extraction process of reflection (col. 9 lines 9-10)" so that the mechanism does not only determine the method parameter types (i.e. the two string objects in the interface class 302, col.9 lines 1-8) but also extracts the actual parameter names to allow the "parameters to be converted to properties of the method bean created from the original (col. 9 lines 1-19)." Accordingly, the generated proxy components (beans) by the extraction process (i.e. reflection) mirror the original components so that the "original components are exposed by the proxy components (i.e. col. 22 lines 46-53)" and "exercised under control of the proxy components (col. 8 lines 11-18 and 23-27)" without being directly accessed or manipulated. Therefore, in Kobayashi, it is also true that "data is collected for evaluation" only without being directly manipulated.

2) Appellants contend that: Kobayashi does not disclose a test case interface mirrored on both sides of a wrapper (brief, page 3, last two lines).

In response, the claims do not use the expression, "mirrored on both sides of a wrapper." The claims recite that the test case to use the wrapper component interface to pass/access the test component and then to generate the test data from the test code in the wrapper component. In Kobayashi, the bean tester (test case) tests the proxy

beans not the actual component (i.e. "once the proxy beans are constructed, they can be displayed, manipulated, and tested by means of a bean tester that is based on conventional visual builder," col. 17 lines 38-41) "within the visual builder by means of the universal transport API 206 which allows the code which implements the underlying objects and components 202 to be exercised under control of the proxy components (i.e. col. 8 lines 19-28)." When the methods of proxy beans are invoked (call) by the bean tester, the methods "invoke the actual component code in order to test the method ...[and] "the method parameters of the original bean are exposed by the proxy components created from the methods of that bean. Consequently, each method can be tested fully (i.e. col. 22 lines 46-53)." This describes a test case interface "mirrored" on both side of a wrapper (tester <-> proxy components <-> actual components (test components)). It is clear that the tester does not directly invoke the actual components and does not output the test result from them. The tester communicates with the proxy components by invoking them and gathering the test result from them to check if the actual components are working properly (i.e. col. 22 lines 66-67 and 41-55).

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Insun Kang

AU 2193

Conferees:

Meng-Ai An (SPE, AU 2193)

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 21

Eddie Lee

EDDIE C. LEE
SUPERVISORY PATENT EXAMINER